Clustered and robust standard errors in Stata and R

Robert McDonald

March 19, 2019

Contents

1 License

2 Introduction

This document illustrates estimation with clustered standard errors in both Stata and R. The pdf version shows both Stata and R output, while the html version shows only R output. The purpose is to illustrate R counterparts to the procedures documented in Petersen (2009) and the companion website, which uses Stata. I do not explain the econometric issues or make any claims about the superiority of an estimator.[1] This is purely an exercise in mimicking results across statistical packages, something that seems to be a Frequently Asked Question.[2]

The commands in Stata and R differ, as the languages differ. In Stata, statistical procedures are a command, with `vce(robust)` and `vce(cluster)` as options. In R, coefficient estimation is a function and the output from this estimation can then be used as the input to another function to compute the covariance matrix. These two components (estimates and covariance matrix) can then be fed to a function that computes standard errors and p-values. For an R user, Stata may seem too inflexible and hard-coded, while for a Stata user, R may seem needlessly complicated.

In what follows, standard Stata calculations are used for single clustering. The function for double clustering, `cluster2`, was written by Mitchell Petersen and obtained from his web site. This was also the source for the other functions with double clustering: logit2.ado, probit2.ado, and tobit2.ado

In all cases, standard and widely-adopted R packages are used to compute both single and double clusters. For examples of clustering with R see the documentation for the sandwich package, especially Berger, Graham, and Zeileis (2017). The approach to computing clustered standard errors is identical in all cases we consider. We obtain coefficient estimates (e.g. using `lm`) and then use `vcovCL` from the `sandwich` package to compute the standard errors. By default, `vcovCL` computes robust standard errors, as does the `robust` option in Stata. Optionally, `vcovCL` can cluster along one or more dimensions. The coefficient and covariance estimates are then fed to `lmtest::testcoef`, which returns the estimates, standard errors, and p-values. This procedure also accommodates bootstrapping, using the `vcovBS` function, but I haven't yet experimented yet with that capability.

Here are examples producing identical estimates and roughly the same output in Stata and R.

Stata:

```
use "data/petersen.dta"
reg y x, robust
reg y x, vce(cluster firm)
```

R:

```
library(sandwich)
library(lmtest)
data(PetersenCL)
reg = lm(y ~ x, data=PetersenCL)  ## estimate the regression
```

[1] Berger, Graham, and Zeileis (2017), a vignette from the `sandwich` package, provides an overview of the topic and has numerous references.

[2] There are numerous posts online comparing results in Stata and R, with some matching and some failing to match results. Examples include this Stackoverflow post, and this Princeton tutorial.

```r
print(coeftest(reg, vcovCL(reg, type='HC1')), digits=6)
print(coeftest(reg, vcovCL(reg, type='HC1', cluster=~firm)), digits=6)
```

The R function `lm` generates OLS estimates. The regression object, `reg`, is then an argument when computing standard errors using `vcovCL`.[3]

This document was created by knitting the Rmarkdown document `stata_and_R_clustering.Rmd`.[4]Tables in Stata were produced using outreg2, and those in R using `stargazer` (Hlavac 2018).

[3] The `vcovCL` function is object-oriented, so it adapts to whatever regression object it receives. One benefit of this approach is that the author of an estimation package can rely on the `sandwich` package for standard errors. Similarly, improvements in the `sandwich` package automatically accrue to estimation packages.

[4] If you have Stata installed, `rmarkdown::render` can execute Stata code and include the output in a document. The restriction is that the `render` function executes each Stata chunk is a separate Stata process, so data is not shared across chunks. This is not a restriction for R or Python chunks.

3 OLS: Vanilla and robust

Here are baseline calculations without clustering and calculating robust standard errors.

3.1 Stata

Results are in Table 1.

```
use "data/petersen.dta"
regress y x
regress y x, robust
```

3.2 R

The variable `reg` contains the full OLS regression output, which is used in subsequent calculations. We save the output from the `coeftest` function. Results are in Table 2.

```
library(sandwich)
library(lmtest)
data(PetersenCL)
reg <- lm(y ~ x, data = PetersenCL) ## `reg` is used throughout
regols = coeftest(reg)  ## OLS
regolsr = coeftest(reg, vcovCL)   ## Robust std errors
```

4 OLS: Cluster by year

Stata results in this section are in Table 1. R results are in Table 2.

4.1 Stata

```
use "data/petersen.dta"
regress y x, vce(cluster year)
```

4.2 R

The `vcovCL` function takes as arguments: the estimated model, the cluster variable, and the type of clustering calculation. We don't need to have separate functions for the covariance matrix and the regression output, but this perhaps illustrates what's going on.

```
v_year = vcovCL(reg, type='HC1', cluster = ~year)
reg_year = coeftest(reg, v_year)
```

Both of the following syntaxes would also work:

```
reg_year1 = coeftest(reg, vcovCL(reg, cluster= ~year, type='HC1'))
reg_year2 = coeftest(reg, vcovCL, cluster= ~year, type='HC1')
all.equal(reg_year1, reg_year2)
[1] TRUE
```

5 OLS: Cluster by firm

Stata results in this section are in Table 1. R results are in Table 2.

5.1 Stata

Results are in Table 1.

```
use "data/petersen.dta"
regress y x, vce(cluster firm)
```

5.2 R

```
v_firm = vcovCL(reg, type="HC1", cluster = ~firm)
reg_firm = coeftest(reg, v_firm)
```

6 OLS: Cluster by both firm and year

Stata results in this section are in Table 1. R results are in Table 2.

6.1 Stata

This uses Petersen's cluster2 function

```
use "data/petersen.dta"
cluster2 y x, fcluster(firm) tcluster(year)
```

6.2 R

The `sandwich` package handles double clustering in the same way as single clustering:

```
v_both = vcovCL(reg, type='HC1', cluster = ~firm+year)
reg_both = coeftest(reg, v_both)
```

7 OLS: Fama-Macbeth

7.1 Stata

This uses Petersen's fm.ado function from his website.

```
use "data/test_data.dta"
tsset firm year
fm y x
```

7.2 R

The `plm` package handles panel models. Results from the Stata Fama-Macbeth function above can be replicated with a standard panel model using a grouped means estimator (the `plm::pmg` function in R).[5]

```r
library(plm)
fpmg <- pmg(y~x, data=PetersenCL, index=c("year","firm")) ##Fama-MacBeth
reg_fm <- coeftest(fpmg)
```

[5] This is discussed in this blog post.

8 Limited dependent variables

The standard Stata and R procedures produce identical answers for logit and tobit. This is also true for probit, provided that the same numerical procedure is used for estimation. Maximum likelihood and glm estimates differ slightly, but Stata and R produce identical results when using the same estimation procedure.

Petersen (2009) includes examples of double clustering for limited dependent models. To illustrate double clustering with limited dependent models, we will create two new variables, `yl`, which equals 1 if $y > 0$ and 0 otherwise, and `ytrunc`, which equals `y` if $y > 0$ and 0 otherwise. I'm not sure how to think about clustering in the context of limited dependent models, but this stackoverflow post suggests that clustered standard errors for binary response models are not necessarily wrong but don't really make sense.[6]

With R, the same procedure used above works with the `glm` function, which handles limited dependent models. We first estimate the regression without a correction for clustering, then we use `vcovCL` to compute cluster-corrected standard errors.

8.1 Logit

Stata results in this section are in Table 3. R results are in Table 4.

8.1.1 Stata

```
use "data/petersen.dta"
gen yl = (y > 0)
logit yl x
logit yl x, vce(robust)
logit yl x, vce(cluster firm)
logit yl x, vce(cluster year)
logit2 yl x, fcluster(firm) tcluster(year)
```

8.1.2 R

We create the limited dependent variables in R:

```
PetersenCL$yl <- (PetersenCL$y > 0)
PetersenCL$ytrunc <- ifelse(PetersenCL$y > 0, PetersenCL$y, 0)
```

```
reg.logit = glm(yl ~ x, data=PetersenCL, family=binomial(link='logit'))
logit0 = coeftest(reg.logit)
logit_robust = coeftest(reg.logit, vcovCL(reg.logit, type='HC0'))
logit_firm = coeftest(reg.logit, vcovCL(reg.logit, type='HC0', cluster=~firm))
logit_year = coeftest(reg.logit, vcovCL(reg.logit, type='HC0', cluster=~year))
logit_both = coeftest(reg.logit, vcovCL(reg.logit, type='HC0', cluster=~year+firm))
```

[6] As I understand it, the point is that robust standard errors are correct if the model is otherwise specified and estimated correctly. However, with a limited dependent model, the initial coefficient estimates depend on the underlying covariance, so it doesn't make sense to estimate the model and then go back and correct the covariance.

8.2 Probit

In both Stata and R, probit estimation can be performed using maximum likelihood or iterated reweighted least squares (IRLS). The different methods give slightly different answers. This causes confusion, as illustrated in this stackoverflow post.

In this section we compare probit estimates obtained using the `probit` functions available in Stata and R, which use maximum likelihood, and those obtained using the glm functions in Stata and R, which (optionally, in the case of Stata) use IRLS. In each case, when the numerical procedure is the same, results are identical between Stata and R.

8.2.1 Generalized linear model: Stata

We do probit estimation using Stata's glm function with the irls option. We have four sets of probit estimates from Stata. We only perform single clustering in this example as the `vce(cluster)` option can only take one variable.

Results are in Table 5.

```
use "data/petersen.dta"
gen yl = (y > 0)
glm yl x, irls family(binomial) link(probit)
glm yl x, irls family(binomial) link(probit) robust
glm yl x, irls family(binomial) link(probit) vce(cluster firm)
glm yl x, irls family(binomial) link(probit) vce(cluster year)
```

8.2.2 Generalized linear model: R

Here we present results for simple robust standard errors, along with both single and double clustering. Results are in Table 6.

```
reg.probit <- glm(yl ~ x, data=PetersenCL, family=binomial(link='probit'))
probitglm0 = coeftest(reg.probit)
probitglm_robust <- coeftest(reg.probit, vcovCL, type='HC0')
probitglm_firm <- coeftest(reg.probit, vcovCL, type='HC0', cluster=~firm)
probitglm_year <- coeftest(reg.probit, vcovCL, type='HC0', cluster=~year)
probitglm_both <- coeftest(reg.probit, vcovCL, type='HC0', cluster=~firm+year)
```

8.2.3 Maximum likelihood: Stata

We obtain identical estimates in Stata and R when using the `sampleSelection::probit` function in R. It is notable that the `sandwich` package works correctly with the `sampleSelection` package. Results are in Table 7.

```
use "data/petersen.dta"
gen yl = (y > 0)
probit yl x
probit yl x, robust
probit yl x, vce(cluster firm)
probit yl x, vce(cluster year)
probit2 yl x, fcluster(firm) tcluster(year)
```

8.2.4 Maximum likelihood: R

In R, a direct maximum likelihood probit function is available in the sampleSelection package. Results are in Table 8.

```
library(sampleSelection)
reg.probit <- probit(yl ~ x, data=PetersenCL)
probitml0 <- coeftest(reg.probit)
probitml_robust <- coeftest(reg.probit, vcovCL)
probitml_firm <- coeftest(reg.probit, vcovCL, type='HC0', cluster=~firm)
probitml_year <- coeftest(reg.probit, vcovCL, type='HC0', cluster=~year)
probitml_both <- coeftest(reg.probit, vcovCL, type='HC0', cluster=~firm+year)
```

8.3 Tobit

8.3.1 Stata

I'm not sure what outreg2 is doing with the tobit output in this example, but the tables are off. Suggestions welcome. Results are in Tables 9 – 11.

```
use "data/petersen.dta"
gen ytrunc = y
replace ytrunc = 0 if y < 0
tobit ytrunc x, ll(0)
tobit ytrunc x, ll(0) vce(cluster firm)
tobit2 ytrunc x, ll(0) fcluster(firm) tcluster(year)
```

8.3.2 R

Results are in Table 12.

```
library(censReg)
reg.tobit <- censReg(ytrunc ~ x, data=PetersenCL, left=0)
tobit0 <- coeftest(reg.tobit)
tobit_robust <- coeftest(reg.tobit, vcovCL, type='HC0')
tobit_firm <- coeftest(reg.tobit, vcovCL, type='HC0', cluster=~firm)
tobit_year <- coeftest(reg.tobit, vcovCL, type='HC0', cluster=~year)
tobit_both <- coeftest(reg.tobit, vcovCL, type='HC0', cluster=~firm+year)
```

9 Summary tables

The following tables summarize the results for Stata and R. The code that produced these results is in the preceding sections. Stata tables were produced using `outreg2` and R tables were produced using `stargazer`. For details, see the Rmarkdown version of this document.

9.1 OLS and Fama-Macbeth

Table 1: Vanilla and clustered standard errors for OLS, and Fama-Macbeth: Stata

	(1)	(2)	(3)	(4)
VARIABLES	OLS	Robust OLS	Cluster: year	Cluster: firm
x	1.03483***	1.03483***	1.03483***	1.03483***
	(0.028583)	(0.028395)	(0.033389)	(0.050596)
Constant	0.029680	0.029680	0.029680	0.029680
	(0.028359)	(0.028361)	(0.023387)	(0.067013)
Observations	5,000	5,000	5,000	5,000
R-squared	0.208	0.208	0.208	0.208

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

Table 2: Vanilla and clustered standard errors for OLS, and Fama-Macbeth: R

	OLS	Robust OLS	Cluster: year	Cluster: firm	Cluster: both	Fama-Macbeth
	(1)	(2)	(3)	(4)	(5)	(6)
x	1.034830***	1.034830***	1.034830***	1.034830***	1.034830***	1.035590***
	(0.028583)	(0.028395)	(0.033389)	(0.050596)	(0.053558)	(0.033342)
Constant	0.029680	0.029680	0.029680	0.029680	0.029680	0.031278
	(0.028359)	(0.028361)	(0.023387)	(0.067013)	(0.065064)	(0.023356)
Cluster:	None	Robust	Firm	Year	Firm+Year	
Observations	5,000	5,000	5,000	5,000	5,000	5,000
R^2	0.207766	0.207766	0.207766	0.207766	0.207766	0.207498
Note:						*p<0.1; **p<0.05; ***p<0.01

9.2 Logit

Table 3: Vanilla and clustered standard errors for Logit: Stata

	(1)	(2)	(3)	(4)	(5)
	Logit	Logit	Logit	Logit	Logit
VARIABLES		robust	cluster: firm	cluster: year	cluster: both
x	0.81189***	0.81189***	0.81189***	0.81189***	0.81189***
	(0.034611)	(0.034256)	(0.052513)	(0.026292)	(0.047701)
Constant	0.035946	0.035946	0.035946	0.035946	0.035946
	(0.030248)	(0.030264)	(0.059913)	(0.028031)	(0.058816)
Observations	5,000	5,000	5,000	5,000	5,000

Standard errors in parentheses

*** p<0.01, ** p<0.05, * p<0.1

Table 4: Vanilla and clustered standard errors for logit: R

	Logit	Robust	Cluster: firm	Cluster: year	Cluster: both
	(1)	(2)	(3)	(4)	(5)
x	0.811890***	0.811890***	0.811890***	0.811890***	0.811890***
	(0.034611)	(0.034256)	(0.052513)	(0.026292)	(0.047701)
Constant	0.035946	0.035946	0.035946	0.035946	0.035946
	(0.030248)	(0.030264)	(0.059913)	(0.028031)	(0.058816)
Cluster:	None	Robust	Firm	Year	Firm+Year
Observations	5,000	5,000	5,000	5,000	5,000

Note: *p<0.1; **p<0.05; ***p<0.01

9.3 Probit

Stata results in this section are in Table 5. R results are in Table 6.

9.3.1 GLM

Table 5: Probit GLM output: Stata

	(1)	(2)	(3)	(4)
	Probit	Probit	Probit	Probit
VARIABLES		robust	cluster: firm	cluster: year
x	0.49662***	0.49662***	0.49662***	0.49662***
	(0.020225)	(0.020141)	(0.030658)	(0.015463)
Constant	0.022424	0.022424	0.022424	0.022424
	(0.018473)	(0.018475)	(0.036582)	(0.016370)
Observations	5,000	5,000	5,000	5,000

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

Table 6: Vanilla and clustered standard errors for GLM probit: R

	Probit	Robust	Cluster: firm	Cluster: year	Cluster: both
	(1)	(2)	(3)	(4)	(5)
x	0.496622***	0.496622***	0.496622***	0.496622***	0.496622***
	(0.020225)	(0.020141)	(0.030658)	(0.015463)	(0.027809)
Constant	0.022424	0.022424	0.022424	0.022424	0.022424
	(0.018473)	(0.018475)	(0.036582)	(0.016370)	(0.035565)
Cluster:	None	Robust	Firm	Year	Firm+Year
Observations	5,000	5,000	5,000	5,000	5,000

Note: *p<0.1; **p<0.05; ***p<0.01

9.3.2 Maximum likelihood

Stata results in this section are in Table 7. R results are in Table 8.

Table 7: Probit ML output: Stata

	(1)	(2)	(3)	(4)	(5)
		Probit	Probit	Probit	Probit
VARIABLES	OLS	robust	cluster: firm	cluster: year	cluster: both
x	0.49662***	0.49662***	0.49662***	0.49662***	0.49662***
	(0.020247)	(0.020184)	(0.030723)	(0.015486)	(0.027863)
Constant	0.022424	0.022424	0.022424	0.022424	0.022424
	(0.018472)	(0.018474)	(0.036580)	(0.016361)	(0.035559)
Observations	5,000	5,000	5,000	5,000	5,000

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

Table 8: Vanilla and clustered standard errors for ML probit: R

	Probit	Robust	Cluster: firm	Cluster: year	Cluster: both
	(1)	(2)	(3)	(4)	(5)
x	0.496622***	0.496622***	0.496622***	0.496622***	0.496622***
	(0.020247)	(0.020184)	(0.030723)	(0.015486)	(0.027863)
Constant	0.022424	0.022424	0.022424	0.022424	0.022424
	(0.018472)	(0.018474)	(0.036580)	(0.016361)	(0.035559)
Cluster:	None	Robust	Firm	Year	Firm+Year
Observations	5,000	5,000	5,000	5,000	5,000

Note: *p<0.1; **p<0.05; ***p<0.01

9.4 Tobit

Table 9: Tobit output: Stata

		(1) Tobit
EQUATION	VARIABLES	
ytrunc	x	1.00267***
		(0.034526)
	Constant	0.036291
		(0.036011)
	/var(e.ytrunc)	4.06710***
		(0.12489)
	Observations	5,000

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

Table 10: Tobit output: Stata

		(1) Tobit
EQUATION	VARIABLES	cluster: firm
ytrunc	x	1.00267***
		(0.055621)
	Constant	0.036291
		(0.072831)
	/var(e.ytrunc)	4.06710***
		(0.23358)
	Observations	5,000

Robust standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

Table 11: Tobit output: Stata

		(1)
		Tobit
EQUATION	VARIABLES	cluster: both
model	x	1.00267***
		(0.059267)
	Constant	0.036291
		(0.070704)
lnsigma	Constant	0.70147***
		(0.029492)
	Observations	5,000

Standard errors in parentheses
*** p<0.01, ** p<0.05, * p<0.1

Table 12: Vanilla and clustered standard errors for tobit: R

	Tobit	Robust	Cluster: year	Cluster: firm	Cluster: both
	(1)	(2)	(3)	(4)	(5)
x	1.002670***	1.002670***	1.002670***	1.002670***	1.002670***
	(0.034526)	(0.034107)	(0.055621)	(0.039777)	(0.059267)
logSigma	0.701465***	0.701465***	0.701465***	0.701465***	0.701465***
	(0.015353)	(0.016150)	(0.028716)	(0.017493)	(0.029492)
Constant	0.036291	0.036291	0.036291	0.036291	0.036291
	(0.036011)	(0.035942)	(0.072831)	(0.031408)	(0.070704)
Cluster:	None	Robust	Firm	Year	Firm+Year
Observations	5,000	5,000	5,000	5,000	5,000

Note: *p<0.1; **p<0.05; ***p<0.01

10 Appendix

10.1 Table creation

In order to focus on the econometric commands, the code chunks displayed in the text hid the commands necessary to produce the Stata output tables. In this section we present the complete code, including the `outreg2` commands used to compute tables. These listings also show the R print statements that produce terminal output similar to Stata's default terminal output.

The procedure for producting formatted tables differs with Stata and R. In Stata, table options are specified along with each regression, and the resulting file is then read into the final document. With R, regression results are saved in a variable and table options are specified when the table is created.

10.1.1 Stata tables

```
## Base OLS and robust
use "data/petersen.dta"
regress y x
outreg2 using tmp, replace  ctitle(OLS) auto(5) tex(fragment)
regress y x, robust
outreg2 using tmp,  ctitle(Robust OLS) auto(5) tex(fragment)

## Cluster by year
use "data/petersen.dta"
regress y x, vce(cluster year)
outreg2 using tmp,  ctitle(Cluster: year) auto(5) tex(fragment)

## cluster by firm
use "data/petersen.dta"
regress y x, vce(cluster firm)
outreg2 using tmp,  ctitle(Cluster: firm) auto(5) tex(fragment)

## cluster by both
use "data/petersen.dta"
cluster2 y x, fcluster(firm) tcluster(year)
outreg2 using tmp,  ctitle(Cluster: both)   auto(5) tex(fragment)

## fama-macbeth
use "data/test_data.dta"
tsset firm year
fm y x
outreg2 using tmp, ctitle(Fama-Macbeth) auto(5) tex(fragment)
```

Because the table options have been specified in each regression, with LaTeX it is only necessary to use `\input{tmp}` to create the formatted Table 1.

10.1.2 R tables

Each set of commands below produces regression coefficients and standard errors, storing the results in a variable (OLS estimates in `reg`, cluster-by-year estimates in `reg_year`, etc.) These commands run the various regressions and compute standard errors, storing each result in a variable. The `print` statements included below create output that resembles the Stata output for a regression. They are not necessary to create the formatted table.

```r
## Base OLS and robust
library(sandwich)
library(lmtest)
data(PetersenCL)
reg <- lm(y ~ x, data = PetersenCL) ## `reg` is used throughout
regols = coeftest(reg)  ## OLS
print(regols, digits=6)
regolsr = coeftest(reg, vcovCL)   ## Robust std errors
print(regolsr, digits=6)

## Cluster by year
v_year = vcovCL(reg, type='HC1', cluster = ~year)
reg_year = coeftest(reg, v_year)
print(reg_year, digits=6)

## cluster by firm
v_firm = vcovCL(reg, type="HC1", cluster = ~firm)
reg_firm = coeftest(reg, v_firm)
print(reg_firm, digits=6)

## cluster by both
v_both = vcovCL(reg, type='HC1', cluster = ~firm+year)
reg_both = coeftest(reg, v_both)
print(reg_both, digits=6)

## fama-macbeth
library(plm)
fpmg <- pmg(y~x, data=PetersenCL, index=c("year","firm")) ##Fama-MacBeth
reg_fm <- coeftest(fpmg)
print(coeftest(fpmg), digits=6)
```

Table 2 is produced by the `stargazer` function. There are other functions that can produce tables from regression objects, including `texreg`, apsrtable, and `pander`.

```
stargazer(OLS=reg, 'OLS robust'=reg,  Year=reg, Firm=reg,
          'Firm+Year'=reg, 'Fama-Macbeth'=fpmg
         ,se=list(regols[,2], regolsr[,2], reg_year[,2], reg_firm[,2],
                 reg_both[,2], reg_fm[,2])
         ,type=output_type
         ,header=FALSE
         ,column.labels=c('OLS', 'Robust OLS', 'Cluster: year',
                         'Cluster: firm', 'Cluster: both', 'Fama-Macbeth')
         ,title='Vanilla and clustered standard errors for OLS, and Fama-Macbeth: R'
         ,no.space=TRUE  ## no blank line between estimates
         ,digits=6
         ,column.sep.width='2pt'
         ,add.lines=list(c('Cluster:', 'None', 'Robust', 'Firm', 'Year', 'Firm+Year'))
         ,dep.var.caption=''
         ,dep.var.labels=''
         ,dep.var.labels.include=FALSE
         ,model.names=FALSE  ## controls "coefficient" string on top
         ,model.numbers=TRUE
         ,omit.stat=c('f', 'ser', 'adj.rsq')
         ,intercept.bottom=TRUE
         ,label='tbl:ols:r'
         ,font.size='footnotesize'
          )
```

References

Berger, Susanne, Nathaniel Graham, and Achim Zeileis. 2017. "Various Versatile Variances: An Object-Oriented Implementation of Clustered Covariances in R." https://cran.r-project.org/web/packages/sandwich/vignettes/sandwich-CL.pdf.

Hlavac, Marek. 2018. "Stargazer: Well-Formatted Regression and Summary Statistics Tables." https://CRAN.R-project.org/package=stargazer.

Petersen, Mitchell A. 2009. "Estimating Standard Errors in Finance Panel Data Sets: Comparing Approaches." *Review of Financial Studies* 22 (1): 435.